Conic Methods for Unconstrained Minimization and Tensor Methods for Nonlinear Equations

Robert B. Schnabel*

CU-CS-227-82 August 1982

University of Colorado at Boulder
DEPARTMENT OF COMPUTER SCIENCE

*This research was supported by ARO contracts DAAG 29-79-C-0023 and DAAG 29-81-K-0108

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **AUG 1982**	2. REPORT TYPE	3. DATES COVERED **00-00-1982 to 00-00-1982**

4. TITLE AND SUBTITLE **Conic Methods for Unconstrained Minimization and Tensor Methods for Nonlinear Equations**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Colorado,Department of Computer Science,Boulder,CO,80309**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Standard methods for nonlinear equations and unconstrained minimization base each iteration on a linear or quadratic model of the objective function respectively. Recently, methods using two generalizations of the standard models have been proposed for these problems. Conic methods for unconstrained minimization use a model that is the ratio of a quadratic function div:ided by the square of a linear function. Tensor methods for nonlinear equations augment the standard linear model with a simple second order term. This paper surveys the research to date on methods. for unconstrained minimization and nonlinear equations that use conic and tensor models. It begins with a brief summary of the standard methods, so that the paper is essentially selfcontained.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**39**	

ANY OPINIONS, FINDINGS, AND CONCLUSIONS OR RECOMMENDATIONS EXPRESSED IN THIS PUBLICATION ARE THOSE OF THE AUTHOR(S) AND DO NOT NECESSARILY REFLECT THE VIEWS OF THE AGENCIES NAMED IN THE ACKNOWLEDGMENTS SECTION.

Abstract

Standard methods for nonlinear equations and unconstrained minimization base each iteration on a linear or quadratic model of the objective function, respectively. Recently, methods using two generalizations of the standard models have been proposed for these problems. Conic methods for unconstrained minimization use a model that is the ratio of a quadratic function divided by the square of a linear function. Tensor methods for nonlinear equations augment the standard linear model with a simple second order term. This paper surveys the research to date on methods for unconstrained minimization and nonlinear equations that use conic and tensor models. It begins with a brief summary of the standard methods, so that the paper is essentially self-contained.

1. Introduction

The two major unconstrained nonlinear algebra problems are the nonlinear equations problem

$$\text{given } F : R^n \to R^n, \quad \text{find } x_* \in R^n \text{ such that } F(x_*)=0 \tag{1.1}$$

where we assume $F \in C^1$, and the unconstrained minimization problem,

$$\underset{x \in R^n}{\text{minimize}} \ f : R^n \to R \tag{1.2}$$

where we assume $f \in C^2$. Computational methods exist that solve many such problems successfully and efficiently, but research aimed at improving these methods continues. In this paper, we discuss two recently introduced classes of algorithms for solving these problems, conic methods for unconstrained minimization and tensor models for nonlinear equations. Both classes contain interesting innovations and seem to offer advantages over the standard methods, although it is too early to access the ultimate importance of either one.

We assume the reader has at least some familiarity with computational methods for nonlinear equations and unconstrained minimization, although we briefly summarize the leading methods in Section 2. Some survey papers on these methods include Brodlie [1977], Dennis [1977], Schnabel [1982a] and More' and Sorensen [1982]. The books by Fletcher [1980], Gill, Murray, and Wright [1981], and Dennis and Schnabel [1983] contain a more detailed treatment.

We will denote the matrix of first partial derivatives of F at x, the Jacobian matrix, by $F'(x) \in R^{n \times n}$; here $F'(x)[i,j]=\partial f_i(x)/\partial x[j]$ where $f_i: R^n \to R$ is the i^{th} component function of $F(x)$. We will denote the vector of first partial derivatives of f at x, the gradient vector, by $\nabla f(x) \in R^n$, and the symmetric matrix of second partial derivatives of f at x, the Hessian matrix, by $\nabla^2 f(x) \in R^{n \times n}$; $\nabla f(x)[i]=\partial f/\partial x[i]$ and $\nabla^2 f(x)[i,j]=\partial^2 f/\partial x[i]\partial x[j]$. Note that we are denoting the i^{th} component of a vector x by $x[i]$ so that we can reserve the notation

x_i for the ith iterate in a sequence of vectors $\{x_k \in R^n\}$.

The main difference between standard methods and conic and tensor methods is in the local model of the nonlinear function that the method uses in determining its iterates. Standard methods for nonlinear equations base the step from the current iterate x_c upon a linear model of $F(x)$ around x_c,

$$M(x_c+d) = F(x_c) + J_c d \tag{1.3}$$

where $d \in R^n$ and $J_c \in R^{n \times n}$ is $F'(x_c)$ or some approximation to it. Similarly, standard methods for unconstrained minimization base each iteration upon a quadratic model of $f(x)$ around x_c,

$$m(x_c+d) = f(x_c) + g_c^T d + \frac{1}{2} d^T H_c d \tag{1.4}$$

where $g_c \in R^n$ is $\nabla f(x_c)$ or a finite difference approximation to it, and $H_c \in R^{n \times n}$ is $\nabla^2 f(x_c)$ or some symmetric approximation to it. These two models are closely related because the minimizer of $f(x)$ must occur at a point x_* where $\nabla f(x_*)=0$, and the gradient of the model (1.4),

$$\nabla m(x_c+d) = \nabla f(x_c) + H_c d \tag{1.5}$$

is a linear model of the system of nonlinear equations $\nabla f(x) : R^n \rightarrow R^n$.

The two new classes of methods are based upon generalizations of (1.3) and (1.4). Conic methods for unconstrained minimization base each step on a model of the form

$$\hat{m}(x_c+d) = f(x_c) + \frac{g_c^T d}{1+b_c^T d} + \frac{\frac{1}{2} d^T A_c d}{(1+b_c^T d)^2} \tag{1.6}$$

where $A_c \in R^{n \times n}$ is symmetric and $b_c \in R^n$. Tensor methods base each iteration on a model of the form

$$\hat{M}(x_c+d) = F(x_c) + J_c d + T_c dd \tag{1.7}$$

where $T_c \in R^{n \times n \times n}$ has a particularly simple form. Here we use the notation $T_c dd$ to denote the vector in R^n whose i^{th} component is

$$(T_c dd)[i] = \sum_{j=1}^{n} \sum_{k=1}^{n} T_c[i,j,k]\cdot d[j]\cdot d[k]. \qquad (1.8)$$

Of course the justification for either of these models is not obvious and we explain it in this paper. Conic models were introduced by Davidon [1980] and also have been investigated by Bjorstad and Nocedal [1979], Sorensen [1980], Stordahl [1980], Davidon [1982], Gourgeon and Nocedal [1982], and Schnabel [1982b]. Tensor models were introduced by Schnabel and Frank [1982] and also are discussed in Frank [1982]. The main goal of the developers of tensor methods is to improve the performance of existing methods on problems were $F'(x_*)$ is singular or ill-conditioned, while at least maintaining the performance of the existing methods on all other problems. The developers of conic methods do not seem to have a similarly limited objective.

The remainder of the paper is organized as follows. Section 2 provides a brief survey of the leading standard methods for nonlinear equations and unconstrained minimization, which are based on the models (1.3) and (1.4) respectively. These include both the derivative methods used when $F'(x)$ or $\nabla^2 f(x)$ are available analytically or from finite differences, and the secant methods that are used otherwise. We concentrate on the ideas and properties that are relevant to our discussion of conic and tensor methods. A reader familiar with these methods should skip or skim Section 2. In Section 3 we briefly discuss several extensions of the standard methods that help motivate conic and tensor methods. These are the methods of Barnes [1965] and Gay and Schnabel [1978] for nonlinear equations and of Davidon [1975] for unconstrained minimization. They all still use the standard models (1.3) and (1.4), but some of their objectives and techniques are similar to conic and tensor methods. We discuss conic methods in Section 4, and tensor methods in Section 5. We comment briefly on the application of these two classes of methods to other nonlinear problems in Section 6.

In our opinion, this paper covers most of the important methods for nonlinear equations and unconstrained minimization based on nonstandard models. There has been occasional other work along these lines, however. Perhaps most significant are the methods for homogeneous functions investigated by Jacobson and Oxsman [1972], Charalambous [1973], Kowalik and Ramakrishnan [1976], and others. These methods do not seem to have led to improved algorithms for general classes of problems.

2. Standard models and methods

The fundamental method for solving the nonlinear equations problem is Newton's method. It consists of choosing the new iterate, x_+, as the root of the linear model of $F(x)$ around x_c,

$$M(x_c+d) = F(x_c) + F'(x_c)d, \tag{2.1}$$

the first two terms of the Taylor series. If $F'(x_c)$ is nonsingular, (2.1) has a unique root at

$$x_+ = x_c - F'(x_c)^{-1}F(x_c). \tag{2.2}$$

If $F(x_*)=0$, $F'(x_*)$ is nonsingular, and $F'(x)$ is Lipschitz continuous in an open neighborhood containing x_*, then the sequence produced by iterating (2.2) is well-defined and converges q-quadratically to x_*, provided the starting point $x_0 \in R^n$ is sufficiently close to x_*. A method that converges provided it is started sufficiently close to the solution is called *locally convergent*. (For our definitions of rates of convergence, see for example Ortega and Rheinboldt [1970] or Dennis and Schnabel [1983].)

There are four weaknesses of Newton's method as a computational procedure for solving systems of nonlinear equations that we wish to discuss. They are

(1) The sequence of iterates may not converge to any root if x_0 is not sufficiently close to a root.

(2) The iteration (2.2) is not well-defined computationally if $F'(x_c)$ is singular or ill-conditioned.

(3) Newton's method usually is slowly locally convergent or does not converge at all to a root where $F'(x_*)$ is singular.

(4) $F'(x)$ may not be available in practical applications.

The first difficulty is addressed by modifying (2.2) when necessary so that the method converges to a root from starting points outside the region of local convergence. This property is called *global convergence*. The most common modifications to achieve global convergence are the line search, where each x_+ is chosen by

$$x_+ = x_c - \lambda_c F'(x_c)^{-1}F(x_c) \tag{2.3}$$

for some $\lambda_c > 0$, and the trust region approach, where x_+ is chosen by

$$x_+ = x_c - (F'(x_c)^T F'(x_c) + \alpha_c I)^{-1} F'(x_c)^T F(x_c) \tag{2.4}$$

with $\alpha_c \geq 0$. In both cases, the real valued parameter λ_c or α_c is selected so that x_+ is a satisfactory next iterate, for example so that $||F(x_+)||_2 < ||F(x_c)||_2$. In the line search, Newton's method corresponds to $\lambda_c = 1$, and it is guaranteed that $||F(x_+)||_2 < ||F(x_+)||_2$ for sufficiently small positive λ_c. In the trust region formula (2.4), Newton's method is $\alpha_c = 0$, and $||F(x_+)||_2 < ||F(x_c)||_2$ is guaranteed for sufficiently large positive α_c. Since the new algorithms use the same types of modifications to achieve global convergence, we do not discuss these strategies further. Many of the references listed in the second paragraph of Section 1 contain information on these strategies.

Various modification may be made to these methods when $F'(x_c)$ is singular or ill-conditioned. These include: i) replacing $F'(x_c)^{-1}$ in the line search formula (2.3) by the pseudo-inverse $F'(x_c)^+$, where the pseudo-inverse of $A \in R^{n \times n}$ may be defined by

$$A^+ = \lim_{\gamma \to 0_+} (A^T A + \gamma I)^{-1} A^T; \qquad (2.5)$$

ii) replacing $F'(x_c)^{-1}$ in the line search by $(F'(x_c)^T F'(x_c) + \gamma I)^{-1} F'(x_c)^T$ with an appropriate small positive value of γ; iii) using the trust region iteration (2.4) with α_c strictly positive. For further information, see Section 6.5 of Dennis and Schnabel [1983]. We mention this difficulty mainly because tensor models for nonlinear equations deal with it nicely.

If $F'(x_*)$ is singular, the convergence of the existing methods to x_* usually is linear at best, even with the above modifications. (See Decker and Kelley [1980a, 1980b, 1982], Griewank [1980], Griewank and Osborne [1981], Reddien [1978, 1980], Rall [1966] for a discussion of the convergence of Newton's method on singular problems.) Some modifications have been proposed to speed convergence on singular problems (see many of the same references), but they mainly require apriori knowledge that $F'(x_c)$ is singular and do not seem suitable for general classes of problems.

If the Jacobian matrix $F'(x)$ is not available in analytic form, it may be approximated by finite differences, meaning that the j^{th} column of $F'(x_c)$ is approximated by

$$(J_c)_{column j} = \frac{F(x_c + he_j) - F(x_c)}{h} \qquad (2.6)$$

for some small $h \in R$. (Here e_j denotes the j^{th} unit vector.) If the expense of this approximation, n additional evaluations of $F(x)$ per iteration, is acceptable, this is done and the aforementioned methods are used with (2.6) in place of $F'(x_c)$. If the stepsizes h are chosen correctly, these is little or no deterioration in performance when changing from analytic to finite difference Jacobians.

If the additional cost of finite difference Jacobian approximation is unacceptable, then a class of methods referred to as *secant* (or quasi-Newton) methods is used instead. These methods replace $F'(x_c)$ in formulas (2.2), (2.5), or (2.6) by a less precise approximation J_c calculated as follows. At the first iteration, J_0 is the finite difference approximation to $F'(x_0)$. After the step from x_c to x_+ is determined, the approximation J_c to $F'(x_c)$ is updated into an approximation J_+ to $F'(x_+)$. The most commonly used updating rule is

$$J_+ = J_c + \frac{(y_c - J_c s_c)s_c^T}{s_c^T s_c} \tag{2.7}$$

where

$$s_c = x_+ - x_c, \quad y_c = F(x_+) - F(x_c). \tag{2.8}$$

This update was introduced by Broyden [1965]. It obeys the *secant equation*

$$J_+ s_c = y_c, \tag{2.9}$$

the multi-dimensional generalization of the standard one dimensional secant equation. For any J_+ that obeys (2.9), the new linear model of $F(x)$ around x_+,

$$\overline{M}(x_+ + d) = F(x_+) + J_+ d \tag{2.10}$$

obeys

$$\overline{M}(x_+) = F(x_+), \quad \overline{M}(x_c) = F(x_c). \tag{2.11}$$

Update (2.7) is selected because of all the matrices obeying (2.9), J_+ given by (2.7) is the closest to J_c in the Frobenius norm. (The Frobenius norm of a matrix or tensor is the square root of the sum of the squares of all the matrix's or tensor's components.)

The local method obtained by using (2.7) to calculate the Jacobian approximations with J_0 a finite difference approximation to $F'(x_0)$, and using

$$x_+ = x_c - J_c^{-1} F(x_c) \tag{2.12}$$

to calculate the steps is referred to as *Broyden's method.* It is locally q-superlinearly convergent to a root x_* under the same assumptions on $F(x)$ and x_* stated above for the q-quadratic convergence of Newton's method. Notice that a secant method for nonlinear equations requires the values of $F(x)$ at the iterates, and no other function or derivative values. In general, secant methods for nonlinear equations or unconstrained minimization usually require more iterations to solve a particular problem than the corresponding analytic or finite difference derivative method, but they usually require fewer function evaluations than the finite difference method. Thus they usually are preferred for problems where function evaluation is expensive and analytic derivatives are unavailable.

The above discussion of secant methods, while cursory, contains the background required for our forthcoming consideration of conic and tensor models. In particular, we emphasize the interpolation property (2.11) that results from formulas (2.9) and (2.10). For further information on these methods, see Dennis and More' [1977], or the references in paragraph 2 of Section 1.

Now let us turn to unconstrained minimization. Newton's method for unconstrained minimization is based on the quadratic model of $f(x)$ around x_c,

$$m(x_c + d) = f(x_c) + \nabla f(x_c)^T d + \tfrac{1}{2} d^T \nabla^2 f(x_c) d, \tag{2.13}$$

the first three terms of the Taylor series. If $\nabla^2 f(x_c)$ is positive definite, $m(x_c+d)$ has a unique minimizer at

$$x_+ = x_c - \nabla^2 f(x_c)^{-1} \nabla f(x_c). \tag{2.14}$$

Alternatively, the iteration (2.14) can be derived by considering the linear model of $\nabla f(x)$ around x_c,

$$\tilde{M}(x_c+d) = \nabla f(x_c) + \nabla^2 f(x_c) d \tag{2.15}$$

and selecting x_+ as the root of $\tilde{M}(x_c+d)$. Viewed in this way, (2.14) is just the application of Newton's method for nonlinear equations to the problem $\nabla f(x)=0$. Therefore it is locally q-quadratically convergent to any point x_* where $\nabla f(x_*)=0$, $\nabla^2 f(x_*)$ is nonsingular, and $\nabla^2 f(x)$ is Lipschitz continuous in an open neighborhood containing x_*. Such a point may be a minimizer, maximizer, or saddle point of $f(x)$.

The four weaknesses of Newton's method for nonlinear equations that we discussed carry over to Newton's method for unconstrained minimization, and the solutions are similar. Global convergence usually is achieved by modifying

(2.14) to

$$x_+ = x_c - \lambda_c H_c^{-1} \nabla f(x_c) \tag{2.16}$$

or

$$x_+ = x_c - (\nabla^2 f(x_c) + \alpha_c I)^{-1} \nabla f(x_c). \tag{2.17}$$

In the first case, $H_c = \nabla^2 f(x_c)$ if $\nabla^2 f(x_c)$ is safely positive definite, otherwise H_c is some positive definite modification of $\nabla^2 f(x_c)$, for example $H_c = \nabla^2 f(x_c) + \gamma I$ with γ large enough to make H_c positive definite. Then it is guaranteed that $f(x_+) < f(x_c)$ for sufficiently small positive λ_c. In the second case, α_c is nonnegative if $\nabla^2 f(x_c)$ is positive definite, and larger than the magnitude of the most negative eigenvalue of $\nabla^2 f(x_c)$ otherwise. It is guaranteed that $f(x_+) < f(x_c)$ for sufficiently large positive α_c. The conic methods we discuss use the same strategies; no further understanding of these strategies is required for the purposes of this paper.

Modifications (2.16) or (2.17) sucessfully deal with the problem of defining a satisfactory step when $\nabla^2 f(x_c)$ is singular or ill-conditioned. However, standard methods still usually converge linearly at best to a point where $\nabla f(x_*)=0$ and $\nabla^2 f(x_*)$ is singular.

Finally, Newton's method for unconstrained minimization requires both the gradient vector $\nabla f(x)$ and the Hessian matrix $\nabla^2 f(x)$. If the gradient is not available analytically, it must be approximated by finite differences since accurate gradient values are essential. If the Hessian matrix is not available, $\nabla^2 f(x)$ is replaced by a finite difference approximation if evaluation of $f(x)$ is inexpensive, by a secant approximation otherwise. Secant approximations for unconstrained minimization are derived similarly to Broyden's update for nonlinear equations. After a step from x_c to x_+, the approximation H_c to $\nabla^2 f(x_c)$ is updated into an approximation H_+ to $\nabla^2 f(x_+)$ obeying

$$H_+ s_c = y_c \tag{2.18}$$

where

$$s_c = x_+ - x_c, \quad y_c = \nabla f(x_+) - \nabla f(x_c). \tag{2.19}$$

Thus the quadratic model

$$\overline{m}(x_+ + d) = f(x_+) + \nabla f(x_+)^T d + \tfrac{1}{2} d^T H_+ d \tag{2.20}$$

satisfies the interpolation conditions

$$\overline{m}(x_+) = f(x_+), \quad \nabla\overline{m}(x_+) = \nabla f(x_+), \quad \nabla\overline{m}(x_c) = \nabla f(x_c). \tag{2.21}$$

In addition, H_+ is chosen to be symmetric since $\nabla^2 f(x)$ always is symmetric. Still, many symmetric H_+ satisifying (2.18) exist; the most used choice is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update

$$H_+ = H_c + \frac{s_c s_c^T}{y_c^T s_c} - \frac{H_c s_c s_c^T H_c}{s_c^T H_c s_c}. \tag{2.22}$$

If H_c is positive definite and

$$s_c^T y_c > 0, \tag{2.23}$$

H_+ is positive definite as well. In practice the initial approximant H_0 is chosen to be positive definite and the step selection strategy enforces (2.23), so all the BFGS approximants to the Hessian are positive definite. This simplifies the modifications required to achieve global convergence. The local method resulting from using (2.22) to define the Hessian approximations and

$$x_+ = x_c - H_c^{-1} \nabla f(x_c) \tag{2.24}$$

to define the steps is locally superlinearly convergent to a point x_* where $f(x)$ and x_* obey the conditions for the q-quadratic convergence of Newton's method, if in addition $\nabla^2 f(x_*)$ is positive definite and H_0 is sufficiently close to $\nabla^2 f(x_0)$.

In summary, we divide the methods considered in this paper into the four categories given in Table 2.1 below. Table 2.1 also lists the information interpolated by the local models at x_c used by the standard methods for each category. We denote the iterate before x_c by x_{prev}.

In Section 4 we also refer to another type of method for unconstrained minimization, *conjugate direction methods*. These methods are related to

Table 2.1 -- Categories of methods considered in this paper

(1) First derivative methods for nonlinear equations
local model at x_c interpolates $F(x_c)$, $F'(x_c)$

(2) Secant methods for nonlinear equations
local model at x_c interpolates $F(x_c)$, $F(x_{prev})$

(3) Second derivative methods for unconstrained minimization
local model at x_c interpolates $f(x_c)$, $\nabla f(x_c)$, $\nabla^2 f(x_c)$

(4) Secant methods for unconstrained minimization
local model at x_c interpolates $f(x_c)$, $\nabla f(x_c)$, $\nabla f(x_{prev})$

secant methods for unconstrained minimization in that they use function and gradient information only. They differ in that do not use any approximation to the Hessian, and require only $O(n)$ storage. Thus they are mainly intended for problems where n is large and the use of $O(n^2)$ storage locations is undesirable. Many of them have the property that if $f(x)$ is a positive definite quadratic, then the n^{th} iterate of the method will be the minimizer x_*. Space does not permit us to describe conjugate direction methods further here; they are described thoroughly in Fletcher [1980], Hestenes [1980], and Gill, Murray, and Wright [1981].

The conic and tensor methods to be described are based on generalizations of the models discussed in this section. As motivation, Table 2.2 summarizes the properties of a good model.

Table 2.2 -- Properties of a good model for nonlinear equations or unconstrained minimization

(1) The model should interpolate useful information.
(2) The model should be a useful approximation to the problem.
(3) The model should be easy to form.
(4) The model should be easy to solve.

Conic and tensor models aim to improve properties 1 and 2 without seriously harming properties 3 and 4. The hope is that the additional costs incurred in items 3 and 4 will be offset by gains in the efficiency or success rate of the algorithm. As a point of reference, Table 2.3 summarizes the costs that may be used to measure the efficiency of algorithms for nonlinear equations or unconstrained minimization, and where applicable, the costs incurred by the standard methods.

Table 2.3 -- Costs of solving nonlinear equations or unconstrained minimization problems by standard methods

(1) Algorithmic overhead (dominated by cost of solving the linear system to find the Newton or secant step; $O(n^3)$ for derivative methods, $O(n^2)$ for secant methods)
(2) Storage (between $n^2/2$ and $2n^2$ locations)
(3) Evaluations of $F(x)$ or $f(x)$, and possibly derivatives

In many practical applications, the evaluations of $F(x)$ or $f(x)$ are very expensive and are the dominant cost. Therefore when assessing the efficiency of new methods, it is desirable that they solve problems using fewer evaluations of the nonlinear function. It also is important, however, that they do not appreciably increase the algorithmic overhead or storage requirements of the standard algorithms.

3. Interpolating additional information using standard models

Conic and tensor models interpolate more function and derivative information than the standard models listed in Table 2.1, by using a more general model. In one of our four problem classes, secant methods for nonlinear equations, it is in fact possible to interpolate additional information using the standard model. For unconstrained minimization, this is only possible in general for quadratic objective functions. We briefly discuss these ideas to motivate further conic and tensor methods.

Secant methods for nonlinear equations use the model

$$M(x_+ + d) = F(x_+) + J_+ d \tag{3.1}$$

to model $F(x)$ around x_+. The secant equation

$$J_+ s_c = y_c \tag{3.2}$$

guarantees $M(x_c) = F(x_c)$. Suppose we also want the model to interpolate the function values $F(x_{-i})$ at some previous iterates x_{-i}, $i=1, \cdots ,p$. This requires

$$F(x_{-i}) = F(x_+) + J_+(x_{-i} - x_+) \tag{3.3}$$

or

$$J_+ s_i = y_i \tag{3.4}$$

where

$$s_i = x_+ - x_{-i}, \quad y_i = F(x_+) - F(x_{-i}). \tag{3.5}$$

Since J_+ is an $n \times n$ matrix, we may satisfy (3.2), plus (3.4) for up to $n-1$ values of i, as long as $s_c, s_1, \cdots, s_{n-1}$ are linearly independent. This possiblity, and a generalization of Broyden's method that achieves it, was first proposed by Barnes [1965]. However the method was not very successful in practice; problems arose when the directions to the past points $s_c, s_1, \cdots, s_{n-1}$ were linearly dependent or close to being dependent. Gay and Schnabel [1978] revived and modified Barnes' idea. By limiting the set of additional past function values to be interpolated to $p<n$ points for which $s_c, s_1, \cdots, s_p$ are strongly linearly

independent, they were able to construct a locally q-superlinearly convergent algorithm that appears quite competitive with a standard Broyden's method algorithm in practice. We do not discuss their method further here. We emphasize, however, the two ideas that will recurr in the forthcoming methods: i) using the model to interpolate function values at previous iterates, and ii) limiting these previous iterates, possibly in a fairly restrictive manner.

The obvious extension of the idea of Barnes and Gay and Schnabel to unconstrained minimization does not work in general. The extension would be to ask the secant model of $f(x)$ around x_+,

$$\nabla m(x_+ + d) = \nabla f(x_+) + H_+ d \tag{3.6}$$

to interpolate $\nabla f(x_{-i})$ at some previous iterates x_{-i} as well as interpolating $\nabla f(x_c)$ at x_c. The difficulty comes from the required symmetry of H_+. Suppose for the model (3.6), $\nabla m(x_c) = \nabla f(x_c)$ and $\nabla m(x_{-1}) = \nabla f(x_{-1})$. This would require

$$H_+ s_c = y_c \quad \text{and} \quad H_+ s_1 = y_1 \tag{3.7}$$

where s_c and s_1 are defined as above and

$$y_c = \nabla f(x_+) - \nabla f(x_c), \quad y_1 = \nabla f(x_+) - \nabla f(x_{-1}). \tag{3.8}$$

Since H_+ is symmetric, (3.7) would imply

$$s_1^T y_c = s_c^T y_1 \tag{3.9}$$

since both sides of (3.9) equal $s_1^T H_+ s_c$. While (3.9) is satisfied if $f(x)$ is a quadratic, it is not satisfied in general for nonquadratic functions.

Davidon [1975] proposed a secant method for unconstrained minimization that interpolates up to n past gradients when $f(x)$ is quadratic, and suggested an extension to general objective functions. Schnabel [1977] studied Davidon's method and proposed several other extensions. None of these have proven superior to the standard secant methods for unconstrained minimization in practice. The desire to interpolate additional previous function or gradient information partially motivates some of the conic methods for unconstrained minimization that we discuss next.

4. Conic methods for unconstrained minimization

The use of conic models in unconstrained minimization algorithms was first proposed by Davidon [1980] and much of the following background material is contained in his paper. A conic function has two equivalent algebraic forms. One is the ratio of a quadratic function divided by the square of a linear function,

$$c(d) = \frac{f + h^T d + \frac{1}{2} d^T B d}{(1+b^T d)^2} \tag{4.1}$$

where $f \in R$, $a,d,h \in R^n$, $B \in R^{n \times n}$. Equation (4.1) is equivalent to

$$c(d) = f + \frac{g^T d}{1+b^T d} + \frac{\frac{1}{2} d^T A d}{(1+b^T d)^2} \tag{4.2}$$

where

$$g = h - fb, \quad A = B - gb^T - bg^T - 2fbb^T. \tag{4.3}$$

The form (4.2) is used for the remainder of this section. The function $c(d)$ is called a conic because its level sets are conic sections, i.e., circles, ellipses, parabolas, or hyperbolas. Figure 1 is an example of a conic function in one variable. Note the discontinuity in $c(d)$; in general, function (4.2) is discontinuous along the $n-1$ dimensional hyperplane $\{d \mid 1+b^T d=0\}$, called the *horizon* of $c(d)$. The vector b is called the *gauge vector.*

The conic function (4.2) is related to the quadratic

$$c(s) = f + g^T s + \frac{1}{2} s^T A s \tag{4.4}$$

by

$$s = \frac{d}{1+b^T d}. \tag{4.5}$$

Equation (4.5) is known as a *collinear scaling* of d; if d and s are related by (4.5)

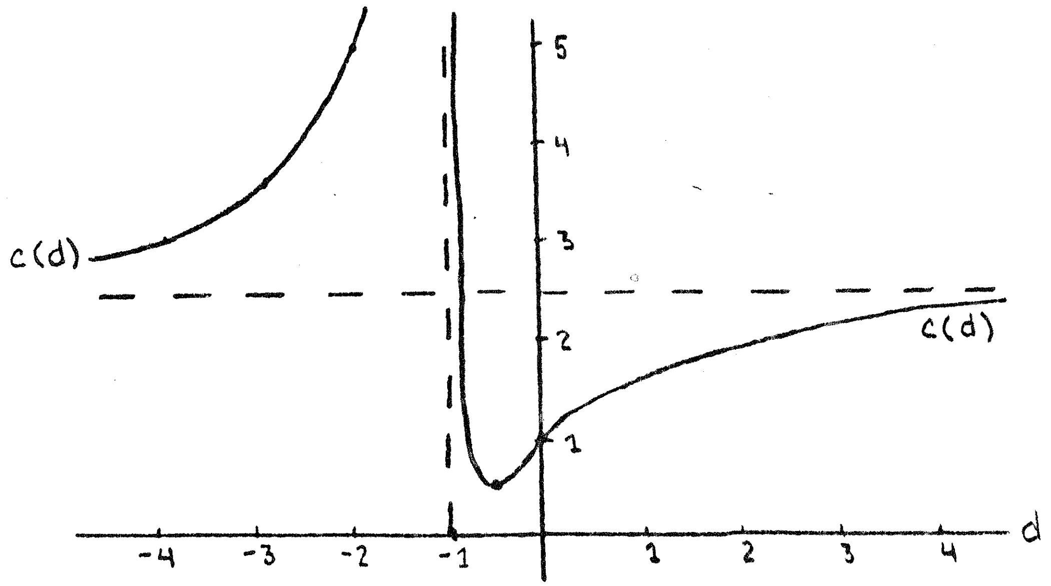

Figure 1 -- The conic function of one variable $c(d) = 1 + \frac{d}{1+d} + \frac{\frac{1}{2}d^2}{(1+d)^2}$

then

$$d = \frac{s}{1-b^T s}. \tag{4.6}$$

Collinear scalings map straight lines to straight lines, affine subspaces to affine subspaces, and most generally, convex sets to convex sets. Furthermore, they are the most general transformation with these properties. (A collinear scaling may have a slightly more general form than (4.5); see Davidon [1980].) Of course, the mapping (4.5) has the discontinuity mentioned above. The relationship between conics and quadratics may be used to derive some of the conic algorithms described in this section, particularly the conjugate direction algorithms of Davidon [1982] and Gourgeon and Nocedal [1982].

To justify the use of the conic function (4.2) in unconstrained minimization algorithms, first we must ask whether a model of this form is appropriate for use in minimization algorithms, and then, to what use the extra degrees of freedom in the model may be put. The answer to the first question is obtained using the same technique that may be used to find the minimzer of a quadratic. Since for nonsingular A, (4.2) can be written

$$c(d) = \tfrac{1}{2}\left[\frac{d}{1+b^Td}+A^{-1}g\right]^T A\left[\frac{d}{1+b^Td}+A^{-1}g\right]+(f-\tfrac{1}{2}g^TA^{-1}g). \qquad (4.7)$$

c(d) has a unique minimer only if A is positive definite. In this case, the minimizer is any d satisfying

$$\frac{d}{1+b^Td}+A^{-1}g=0 \qquad (4.8)$$

if (4.8) has a solution. From (4.6), the solution to (4.8) is

$$d=\frac{-A^{-1}g}{1+b^TA^{-1}g} \qquad (4.9)$$

as long as $1+b^TA^{-1}g\neq 0$. So roughly speaking, a conic model has a unique minimizer in almost the same cases as a quadratic does, that is, when the matrix in the model is positive definite.

To use a conic function of form (4.2) as a model in a minimization algorithm, presumably it should interpolate some function and derivative values of $f(x)$. The first two derivatives of $c(d)$ are

$$\nabla c(d)=\frac{1}{1+b^Td}\left[I-\frac{bd^T}{1+b^Td}\right]\left[g+\frac{Ad}{1+b^Td}\right], \qquad (4.10)$$

$$\nabla^2 c(d)=\frac{A-bg^T-gb^T}{(1+b^Td)^2}-\frac{2Agb^T+2bg^TA}{(1+b^Td)^3}-\frac{bb^T}{(1+b^Td)^3}\left[2g^Td+\frac{3d^TAd}{1+b^Td}\right]. \qquad (4.11)$$

Thus

$$\nabla c(0)=g, \quad \nabla^2 c(0)=A-bg^T-gb^T. \qquad (4.12)$$

Therefore to model $f(x)$ around x_c with a conic model, one should use a model of the form

$$\hat{m}(x_c+d) = f(x_c) + \frac{\nabla f(x_c)^T d}{1+b_c^T d} + \frac{\frac{1}{2} d^T A_c d}{(1+b_c^T d)^2} \qquad (4.13)$$

since it satisfies

$$\hat{m}(x_c) = f(x_c), \quad \nabla \hat{m}(x_c) = \nabla f(x_c), \qquad (4.14)$$

for any $b_c \in R^n$ and $A_c \in R^{n \times n}$. From (4.12), the second derivative matrix of this model is

$$\nabla^2 \hat{m}(x_c) = A_c - b_c \nabla f(x_c)^T - \nabla f(x_c) b_c^T. \qquad (4.15)$$

Most of the research on conic methods is based on a model of form (4.13). If A_c is positive definite and $1+b_c^T A_c^{-1} \nabla f(x_c) \neq 0$, it has a unique minimizer at

$$x_c - \frac{A_c^{-1} \nabla f(x_c)}{1+b_c^T A_c^{-1} \nabla f(x_c)}; \qquad (4.16)$$

the minimizer is on the same side of the horizon as x_c if and only if $1+b_c^T A_c^{-1} \nabla f(x_c) > 0$. In the remainder of this section, we describe how several authors have used conic models in unconstrained minimization algorithms. This includes the choice of the parameters A_c and b_c in (4.13). It is important to note that the matrix A_c wil be different than the matrix used in standard methods for unconstrained minimization. Thus the direction as well as the magnitude of the step to the minimizer of the conic model will be different.

Davidon [1980] and Sorensen [1980] consider the use of a conic model in a secant method for unconstrained minimization. As in our discussion of secant methods in Section 2, assume we have taken a step from x_c to x_+, and are constructing a new conic model of $f(x)$ around x_+,

$$\hat{m}(x_+ + d) = f(x_+) + \frac{\nabla f(x_+)^T d}{1+b_+^T d} + \frac{\frac{1}{2} d^T A_+ d}{(1+b_+^T d)^2}. \qquad (4.17)$$

Davidon and Sorensen show how to use this model to obtain

$$\hat{m}(x_c) = f(x_c), \quad \nabla \hat{m}(x_c) = \nabla f(x_c). \qquad (4.18)$$

as well as $\hat{m}(x_+) = f(x_+)$ and $\nabla \hat{m}(x_+) = \nabla f(x_+)$ which are satisfied for all values of b_+ and A_+. Let $s_c = x_+ - x_c$. The first interpolation condition in (4.18) requires

$$f(x_c) = f(x_+) - \frac{\nabla f(x_+)^T s_c}{1-b_+^T s_c} + \frac{\frac{1}{2} s_c^T A_+ s_c}{(1-b_+^T s_c)^2} \tag{4.19}$$

while the second is satisfied if

$$\nabla f(x_c) = \frac{1}{1-b_+^T s_c}\left[I + \frac{b_+ s_c^T}{1-b_+^T s_c}\right]\left[\nabla f(x_+) - \frac{A s_c}{1-b_+^T s_c}\right]. \tag{4.20}$$

Let $\beta = 1-b_+^T s_c$. Taking the inner product of (4.20) with s_c and then using (4.19) to eliminate the $s_c^T A_+ s_c$ term yields

$$\beta\,[\beta^2 s_c^T \nabla f(x_c) - 2\beta(f(x_+)-f(x_c)) + s_c^T \nabla f(x_+)] = 0. \tag{4.21}$$

If (4.21) has a nonzero real solution (this can be assured by the choice of λ_c in a line search algorithm), then any b_+ that satisifies

$$b_+^T s_c = 1 - \beta \tag{4.22}$$

together with any A_+ that satisfies

$$A_+ s_c = \beta \nabla f(x_+) - \beta^2 \nabla f(x_c) + \beta^2 b_+ s_c^T \nabla f(x_c) \tag{4.23}$$

causes the model (4.17) to satisfy (4.18).

Thus by using (4.21), (4.22), and (4.23), the conic model (4.17) satisfies the three interpolation conditions (2.21) satisfied by standard secant methods for unconstrained minimization, plus $\hat{m}(x_c) = f(x_c)$. This causes the step to the minimizer of the conic model to depend on function as well as gradient values, whereas in a standard secant method, the minimizer of the model is determined solely by gradient values. This fact is cited by several authors as a reason why algorithms based on conic models should be more efficient than the corresponding algorithms based on quadratic models.

Sorensen [1980] proves the local q-superlinear convergence of an algorithm of the type we have just described. His algorithm assures that all the matrices A_+ are positive definite through the choice of the line search parameter, the proper choice from among the two nonzero roots of (4.21), and by using BFGS updates. The parameter b_+ is chosen in the direction $\nabla f(x_c)$. Sorensen also presents some promising test results, but they are from a very limited set of tests, and considerably more testing would be required to establish the utility of this approach.

Davidon [1980] also shows how A_+ and b_+ may be chosen so that the model satisfies the additional interpolation conditions

$$\hat{m}(x_{-i}) = f(x_{-i}) \quad \text{and} \quad \nabla\hat{m}(x_{-i}) = \nabla f(x_{-i}), \quad i=1, \cdots, n-1 \tag{4.24}$$

when $f(x)$ is a conic function, where as in Section 3 $\{x_{-i}\}$ are past iterates. For nonconic functions this is not usually possible, however, and to our knowledge, this idea has not yet been used to make a general purpose unconstrained minimization algorithm.

Schnabel [1982b] and Stordahl [1980] study the use of the model (4.13) when the Hessian matrix is available analytically or by finite differences. The model they use is

$$\hat{m}(x_c+d) = \\ f(x_c) + \frac{\nabla f(x_c)^T d}{1+b_c^T d} + \frac{\frac{1}{2} d^T(\nabla^2 f(x_c) + b_c \nabla f(x_c)^T + \nabla f(x_c) b_c^T) d}{(1+b_c^T d)^2}. \tag{4.25}$$

From (4.14) and (4.15), (4.25) satisfies

$$\hat{m}(x_c) = f(x_c), \quad \nabla\hat{m}(x_c) = \nabla f(x_c), \quad \nabla^2\hat{m}(x_c) = \nabla^2 f(x_c) \tag{4.26}$$

for any $b_c \in R^n$. The vector b_c then may be chosen to allow the model to interpolate additional information. Schnabel and Stordahl impose the requirement

$$\hat{m}(x_{-i}) = f(x_{-i}), i=1, \cdots, p \tag{4.27}$$

at $p \leq n$ past iterates x_{-i}. Substitution into (4.25) shows that this model satisfies $\hat{m}(x_{-i}) = f(x_{-i})$ if

$$b_c^T s_i = \sigma_i = \\ 1 + \frac{\nabla f(x_c)^T s_i - \sqrt{(\nabla f(x_c)^T s_i)^2 + (f(x_{-i}) - f(x_c))(\frac{1}{2} s_i^T \nabla f(x_c) s_i + \nabla f(x_c)^T s_i)}}{f(x_{-i}) - f(x_c)} \tag{4.28}$$

where

$$s_i = x_c - x_{-i}. \tag{4.29}$$

The term inside the square root in (4.28) may be negative but rarely is in practice. Since (4.28) only determines the projection of b_c in the direction s_i, the conic model (4.25) may interpolate up to n past function values $f(x_{-i})$ as long as the directions to the past points, s_i, are linearly independent. If $f(x)$ is quadratic, the right hand side of (4.28) is zero so the choice $b_c = 0$ is allowed.

Schnabel and Stordahl tested an algorithm based on the above model. They found it advantageous to limit the number of past function values interpolated at any iteration to $p \le \sqrt{n}$, and to require the directions s_i to the past points to be strongly linearly independent of each other; roughly speaking, each direction s_i that is used must make an angle of at least 45 degrees with the linear subspace spanned by the other directions $\{s_k\}$ that are used. They chose b_c to be the minimum l_2 norm solution to the underdetermined system of equations

$$b_c^T s_i = \sigma_i, \quad i=1, \cdots, p. \tag{4.30}$$

Schnabel and Stordahl compared their algorithm to an algorithm that used the standard second derivative quadratic model but was identical in all other respects, on the unconstrained minimization test problems in More', Garbow, and Hillstrom [1981]. Each algorithm used the strategy in Dennis and Schnabel [1983] to augment $\nabla f(x_c)$ or A_c to be positive definite when necessary, and then chose the next iterate using a line search. Out of 32 test runs, the conic algorithm was more efficient (in iterations and function evaluations) on 19, less efficient on 11, and tied on 4; on the average, the conic algorithm required 18% fewer iterations and 21% fewer function evaluations than the quadrtic algorithm. It is not clear whether these results justify the added complexity of the conic method.

Another possibility in using model (4.25) is to choose b_c so that $\nabla \hat{m}(x_{-1}) = \nabla f(x_{-1})$ where x_{-1} is the most recent past iterate. This is more in keeping with the philosophy of the algorithms of Davidon and Gourgeon and Nocedal discussed next, as it implies that $\hat{m}(x) = f(x)$ if $f(x)$ is a conic function. We currently are investigating this approach.

Davidon [1982] and Gourgeon and Nocedal [1982] have proposed a conjugate direction method based on conic models. It is a generalization of the standard conjugate gradient method for minimizing quadratics (Hestenes and Stiefel [1952]) and finds the minimizer of a conic function in at most n iterations. An extension of these methods to nonconic functions has not yet been developed, however, so no comparison with existing conjugate direction methods for unconstrained minimization is possible.

Since we do not include a thorough treatment of conjugate direction methods in this paper, we just give a brief description of Davidon's and Gourgeon and Nocedal's algorithms. The key feature is that they show that if $f(x)$ is a conic function, then given the values of $f(x)$ and $\nabla f(x)$ at any three collinear points, the value of the gauge vector b can be determined. The value of the gauge vector changes as the conic is expanded around different points, but only by scalar multiples that Davidon and Gourgeon and Nocedal also show how to calculate. Using this information, they are able to generalize the conjugate gradient algorithm to exactly minimize the conic function along a line d_k at the k^{th} iteration, while also choosing d_k so that the next iterate x_{k+1} is the minimizer of the conic in the affine subspace spanned by the $k-1$ previous step directions d_1, $\cdots$, d_{k-1} as well. Thus the algorithm minimizes a conic function in n or fewer iterations, without storing or using an $n \times n$ matrix. Gourgeon and Nocedal point out several possible numerical instabilities in implementing this algorithm, and seem to remedy them satisfactorily.

Davidon [1982] also proposes a related algorithm that again minimizes a conic function in n or fewer iterations, and also accumulates an approximation to the matrix A in (4.2) as it proceeds. If $f(x)$ is conic, after n iterations this approximation is $A = \nabla^2 f(x_*)$. Gourgeon and Nocedal state that the sequence of points generated by their algorithm and Davidon's two algorithms is identical for conic functions, if exact arithmetic is used. Davidon's secant algorithm probably could be extended into a new secant conic algorithm for unconstrained minimization. This extension would require the evaluation of $f(x)$ and $\nabla f(x)$ at three collinear points on some or all iterations, as would an extension of the conjugate direction algorithms of Davidon or Gourgeon and Nocedal to nonconics.

Finally, Bjorstad and Nocedal [1979] show that the secant conic algorithm originally proposed by Davidon converges quadratically under reasonable assumptions when applied to one dimensional problems. This rate is faster than the order $(1+\sqrt{5}/2) \cong 1.61$ convergence rate of the secant method on one dimensional problems. Similarly, the second derivative conic method of Schnabel and Stordahl converges with order $1+\sqrt{2} \cong 2.41$ on one dimensional problems, compared to the quadratic convergence of Newton's method. These results probably have practical importance only if they extend to multiple dimensions, which is very unlikely.

5. Tensor methods for nonlinear equations

Tensor models for nonlinear equations augment the standard linear model of $F(x)$ around x_c by a second order term. The most obvious tensor model is the first three terms of the Taylor series,

$$\overline{M}(x_c+d) = F(x_c) + F'(x_c)d + \tfrac{1}{2}F''(x_c)dd, \qquad (5.1)$$

where $F'' \in R^{n \times n \times n}$. However the use of model (5.1) in a nonlinear equations algorithm would violate many of the principles expressed at the end of Section 2. In particular, the model would require at least $O(n^3)$ additional operations to form and roughly $n^3/2$ additional locations to store, and finding a root of the model would require solving a system of n quadratic equations at n unknowns at each iteration. Any of these costs clearly is unacceptable.

Instead, Schnabel and Frank [1982] have proposed the use of a model of the form

$$\hat{M}(x_c+d) = F(x_c) + F'(x_c)d + \tfrac{1}{2}T_c dd \tag{5.2}$$

where $T_c \in R^{n \times n \times n}$ has a particulary simple form. In particular, the additional costs of forming, storing, and finding a root of the tensor model all are small in comparison to the corresponding costs already required by Newton's method. In this section, we summarize how Schnabel and Frank determine the term T_c in (5.2), how the resultant model is solved efficiently, and how a nonlinear equations method utilizes this method. Finally, we summarize some of their test results.

The main motivation for the work of Schnabel and Frank was to construct a general purpose method that would be more efficient than standard methods on problems where $F'(x_*)$ is singular or ill-conditioned, and at least as efficient as standard methods on all other problems. An early version of their work is reported in Frank [1982]. For practical purposes, it probably would be more desirable to have a secant tensor method that does not require values of $F'(x)$; we currently are working on such an extension.

To determine the tensor term T_c in (5.2), we require the model $\hat{M}(x_c+d)$ to interpolate the function values $F(x_{-i})$ at $p \le \sqrt{n}$ previous iterates $x_{-1}, \cdots, x_{-p}$. Substituted into (5.2), this requirement is

$$F(x_{-i}) = F(x_c) - F'(x_c)s_i + \tfrac{1}{2}T_c s_i s_i, \quad i=1, \cdots ,p \le \sqrt{n} \tag{5.3}$$

where

$$s_i = x_c - x_{-i}. \tag{5.4}$$

The upper bound of $\sqrt{n}$ past points was suggested by the computational experience of Schnabel and Stordahl mentioned in Section 4 for conic methods that use past function values; it also is required to keep the costs of forming and solving the tensor model small. The past points are selected by the same method as is used in Schnabel and Stordahl's conic code. At each iteration, the most recent past point is selected. Then, for $i=2, \cdots, \sqrt{n}$, the i^{th} past point is selected if it makes an angle of at least 45 degrees with the affine subspace spanned by the already selected subset of the 1^{st} through $i-1^{st}$ past points.

The $p \leq \sqrt{n}$ interpolation conditions (5.3) result in np linear equations in the n^3 elements of T_c, meaning that T_c is underdetermined. Following successful precedent in determining secant updates for nonlinear equations (see e.g. Dennis and Schnabel [1979]), Schnabel and Frank choose the T_c that solves

$$\underset{T_c \in R^{n \times n \times n}}{\text{minimize}} \; ||T_c||_F \tag{5.5}$$

$$\text{subject to} \;\; T_c s_i s_i = 2(F(x_{-i}) - F(x_c) + F'(x_c) s_i), \;\; i=1, \cdots, p$$

where $||\cdot||_F$ denotes the Frobenius norm. They show that the solution to (5.5) has the form

$$T_c = \sum_{i=1}^{p} a_i \; s_i \; s_i \tag{5.6}$$

where $a_i \in R^n$, $i=1, \cdots, p$ and $a_i \; s_i \; s_i$ denotes the rank one tensor whose i,j,k element is $a_i[i] \cdot s_i[j] \cdot s_i[k]$. Thus T_c is a rank p tensor. The p vectors $a_i \in R^n$ are calculated by solving one symmetric and positive definite $p \times p$ system of linear equations with n different right hand sides, requiring a total of $np^2 \leq n^2$ each multiplications and additions.

Given the form (5.6) of T_c, the tensor model (5.2) becomes

$$\hat{M}(x_c + d) = F(x_c) + F'(x_c) d + \sum_{i=1}^{p} a_i \; (s_i^T d)^2 . \tag{5.7}$$

Therefore a maximum of $2n^{1.5}$ additional storage locations are required for the tensor term, to store the p a_i and s_i vectors. (The x_{-i} and s_i vectors can share storage.) The dominant arithmetic cost in forming the tensor term is the $n^2p \leq n^{2.5}$ multiplications and additions required to form the p right hand sides of (5.5). Neither of these costs is significant in comparison to the n^2 locations and $O(n^3)$ arithmetic operations required to store and solve the standard linear model.

To use $\hat{M}(x_c+d)$ given by (5.7) in an algorithm for nonlinear equations, presumably we need to be able to find a $d_* \in R^n$ for which $\hat{M}(x_c+d_*)=0$. However the model may not always have a real root, as is obvious by considering the one dimensional case, and in this case it seems reasonable to find a d_* that minimizes $\|\hat{M}(x_c+d)\|_2$. Thus in general we need an efficient procedure for solving

$$\underset{d_* \in R^n}{\text{minimize}} \ \|\hat{M}(x_c+d)\|_2 \tag{5.8}$$

for $\hat{M}(x_c+d)$ given by (5.7). Schnabel and Frank show how to reduce (5.8) to a much smaller problem of the form

$$\underset{z \in R^p}{\text{minimize}} \ \|Q(z)\|_2, \ \ Q: R^p \rightarrow R^q \tag{5.9}$$

where the q equations in p unknowns $Q(z)$ are quadratic and $p \leq q \leq n$, with usually $q=p$. The reduction is accomplished using orthogonal transformations of the variable and function spaces. It requires the QR decomposition of $F'(x_c)$ which also usually is used in a Newton's method algorithm and takes $2n^3/3$ each multiplications and additions; the next leading term in the reduction is $2n^2p \leq 2n^{2.5}$ additional multiplications and additions which is insignificant in comparison.

Now to find a root or minimizer of the tensor model (5.7) we still must solve the nonlinear problem (5.9); however, this is inexpensive due to the reduced number of variables in (5.9). To solve (5.9) by a nonlinear least squares algorithm costs $O(p^2q)$ arithmetic operations per iteration, and in practice a maximum of $2p$ iterations suffices. Thus the total cost of solving (5.9) is $O(p^3q)$

operations, which is $\leq O(n^2)$ in the normal case when $p=q$, and $\leq O(n^{2.5})$ in all cases. Given the solution to (5.9), the solution to (5.8) is obtained by backsolving and requires $O(n^2)$ arithmetic operations. Therefore the total cost of finding a root or minimizer of the tensor model is not significantly more than calculating the root of the standard linear model using a QR factorization. It is twice as expensive as calculating the root of the standard linear model using Gaussian elimination; however many production codes do use the QR factorization because it facilitates the global portion of the algorithm and the modifications required when $F'(x_c)$ is ill-conditioned (see Dennis and Schnabel [1983]).

An important point is that the model (5.7) may have isolated solutions even when $F'(x_c)$ is singular and $F(x_c)$ is not contained in the subspace spanned by the columns of $F'(x_c)$; if so the method of Schnabel and Frank should find a solution. Similarly, (5.8) often is not an ill-conditioned problem when the solution of the linear model would be. It is hoped that these properties are beneficial on problems where $F'(x_*)$ is singular. Another interesting issue is that (5.7) usually has multiple roots or minimizers; we try to find the d_* closest to the Newton step, by appropriately choosing the starting point $z_0 \in R^n$ in the algorithm that solves (5.9).

Schnabel and Frank have tested an algorithm based on a preliminary version of the above techniques. At each iteration, after determining the solution d_* to (5.8), it takes $x_* = x_c + d_*$ if this is an acceptable next iterate. If not, it does a line search in the Newton direction if d_* is not a descent direction, or occasionally if it prefers the Newton direction for other reasons given in Frank [1982], otherwise it does a line search in the direction d_*. Schnabel and Frank compare their algorithm to an algorithm that uses the standard linear model exclusively but otherwise is identical to the tensor algorithm, using the test problems in More', Garbow, and Hillstrom [1981]. For all but one of these problems (Powell's singular function) $F'(x_*)$ is nonsingular, so they also construct singular versions of the problems in More', Garbow, and Hillstrom, as described in Schnabel and Frank [1982]. The dimensions of the problems range from 2 to 50, with most of the problems having dimension between 10 and 50.

Detailed results of these tests are given in Frank [1982] and in Schnabel and Frank [1982]. In summary, on 44 problems where $F'(x_*)$ is nonsingular, the tensor method requires fewer iterations and function and Jacobian evaluations than the standard method in 30 cases, the same number in 13 cases, and more in 1 case. The improvements rarely are by more than 20%, however most of these problems are quite easy when using analytic Jacobians, requiring 10 or fewer iterations. On a set of 37 problems where $F'(x_*)$ has rank $n-1$, the tensor method requires fewer iterations and function and Jacobian evaluations in 26 cases, the same number in 8 cases, and more in 3 cases. Here the improvement by the tensor method is far more dramatic; the standard method fails to solve 11 of the 37 problems in 150 iterations, while the tensor method solves all of these in at most 27 iterations. On a second differently constructed set of 25 problems where $F'(x_*)$ again has rank $n-1$, the tensor model requires fewer iterations and function and Jacobian evaluations in all cases. The standard method solves 24 of these; on these problems, the tensor algorithm requires an average of 53% of the iterations and 58% of the function evaluations used by the standard algorithm. Finally, on 37 problems where $F'(x_*)$ has rank $n-2$, the tensor model again does better in 26 cases, the same in 8, and worse in 3, again with substantial improvements in many cases. Clearly these results show promise for the tensor methods.

6. Future applications of conic and tensor models

Methods that use conic and tensor models are still in their infancy. Depending upon their success, conic and tensor models might eventually be used in many areas of optimization besides the ones described in Sections 4 and 5. In this section we comment briefly on some of these possibilities.

Conic models or collinear scalings do not seem suited to the nonlinear equations problem. The collinear transformation of the standard linear model for nonlinear equations,

$$\tilde{M}(x_c+d) = F(x_c) + \frac{F'(x_c)d}{1+b_c^Td} \tag{6.1}$$

suffers from the fact that its root is in the Newton direction $-F'(x_c)^{-1}F(x_c)$ for all values of b_c; thus, only a steplength parameter is introduced. On the other hand, conic models certainly could be applied to nonlinear least squares or constrained optimization problems. They also may be useful for solving special classes of optimization problems, for example penalty functions where the horizon of the conic function might help reflect the shape of the penalty function.

Hopefully, tensor models can be applied to all four problem classes listed in Section 2, derivative and secant methods for both nonlinear equations and unconstrained minimization. In fact, our own main practical interest would be in a secant tensor method that quickly solves unconstrained minimization problems with $\nabla^2 f(x_*)$ singular or ill-conditioned, because we see many such problems in practice. An example is overparameterized data fitting problems, where in our experience the objective function usually is sufficiently expensive to evaluate that secant methods are preferred. The extension of the method of Section 5 to a secant method for nonlinear equations, and the extension to unconstrained minimization, both appear to present challenging problems. The tensor method of Section 5 generalizes virtually without change into a Gauss-

Newton or Levenberg-Marquardt type method for solving the nonlinear least squares problem, and since the analytic or finite difference Jacobian always is used in this setting, such a method would be of practical interest. Tensor methods also may be useful in finding complex solutions to nonlinear equations or optimization problems.

Finally, there certainly are other nonstandard models besides conics and tensors that could be considered for nonlinear equations or optimization problems. The main contribution of conic and tensor models may be that they cause researchers to consider various nonstandard models, and that some of these prove to be useful in practice.

Acknowledgements

The impetus for much of the work on nonstandard models for unconstrained optimization has come from the work of William Davidon, especially his papers of 1975 and 1980. I am pleased to acknowledge my debt to Davidon and his ideas; Davidon's 1975 paper on projected updates for unconstrained minimization led to the paper of Gay and Schnabel, Davidon's 1980 paper on conics led to my work on conics and indirectly supplied the stimulus for the work on tensor models.

I also thank Richard Byrd, Paul Frank, Andreas Griewank, and Kari Stordahl for many helpful discussions on these subjects.

7. References

J. Barnes [1965], "An algorithm for solving nonlinear equations based on the secant method", *Computer Journal* 8, pp. 66-72.

P. Bjorstad and J. Nocedal [1979], "Analysis of a new algorithm for one-dimensional minimization", *Computing* 22, pp. 93-100.

K. W. Brodlie [1977], "Unconstrained minimization", in *The State of the Art in Numerical Analysis*, D. Jacobs, ed., Academic Press, London, pp. 229-268.

C. G. Broyden [1965], "A class of methods for solving nonlinear simultaneous equations", *Mathematics of Computation* 19, pp. 577-593.

C. Charalambous [1973], "Unconstrained optimization based on homogeneous models", *Mathematical Programming* 5, pp. 189-198.

W. C. Davidon [1975], "Optimally conditioned optimization algorithms without line searches", *Mathematical Programming* 9, pp. 1-30.

W. C. Davidon [1980], "Conic approximations and collinear scalings for optimizers", *SIAM Journal on Numerical Analysis* 17, pp. 268-281.

W. C. Davidon [1982], "Conjugate directions for conic functions", in *Nonlinear Optimization 1981*, M. J. D. Powell, ed., Academic Press, London, pp. 23-28.

D. W. Decker and C. T. Kelley [1980a], "Newton's method at singular points I", *SIAM Journal on Numerical Analysis* 17, pp. 66-70.

D. W. Decker and C. T. Kelley [1980b], "Newton's method at singular points II", *SIAM Journal on Numerical Analysis* 17, pp. 465-471.

D. W. Decker and C. T. Kelley [1982], "Convergence acceleration for Newton's method at singular points", *SIAM Journal on Numerical Analysis* 19, pp. 219-229.

J. E. Dennis Jr. [1977], "Nonlinear least squares and equations", in *The State of the Art in Numerical Analysis*, D. Jacobs, ed., Academic Press, London, pp. 269-312.

J. E. Dennis Jr. and J. J. More' [1977], "Quasi-Newton methods, motivation and theory", *SIAM Review* 19, pp. 46-89.

J. E. Dennis Jr. and R. B. Schnabel [1979], "Least change secant updates for quasi-Newton methods", *SIAM Review* 21, pp. 443-459.

J. E. Dennis Jr. and R. B. Schnabel [1983], *Numerical Methods for Nonlinear Equations and Unconstrained Optimization*, Prentice-Hall, Englewood Cliffs, New Jersey.

R. Fletcher [1980], *Practical Method of Optimization, Vol 1, Unconstrained Optimization*, John Wiley and Sons, New York.

P. Frank [1982], "A second-order local model for solution of systems of nonlinear equations", M.S. Thesis, Department of Computer Science, University of Colorado at Boulder.

D. M. Gay and R. B. Schnabel [1978], "Solving systems of nonlinear equations by Broyden's method with projected updates", in *Nonlinear Programming 3*, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, pp. 245-281.

P. E. Gill, W. Murray, and M. H. Wright [1981], *Practical Optimization*, Academic Press, London.

H. Gourgeon and J. Nocedal [1982], "A conic algorithm for optimization", preprint.

A. O. Griewank [1980], "Analysis and modification of Newton's method at singularities", Ph.D. thesis, Australian National University.

A. O. Griewank and M. R. Osborne [1981], "Newton's method for singular problems when the dimension of the null space is >1", *SIAM Journal on Numerical Analysis* 18, pp. 145-150.

M. R. Hestenes [1980], *Conjugate-Direction Methods in Optimization*, Springer-Verlag, New York.

M. R. Hestenes and E. Stiefel [1952], "Methods of conjugate gradients for solving linear systems", *Journal of Research of the National Bureau of Standards* 49, pp. 409-436.

D. Jacobsman and W. Oxman [1972], "An algorithms that minimizes homogeneous functions of n variables in n+2 iterations and rapidly minimizes general functions", *Journal of Mathematical Analysis and Applications* 38, pp. 533-552.

J. Kowalik and K. Ramakrishnan [1976], "A numerically stable optimization method based on a homogeneous function", *Mathematical Programming* 11, pp. 50-66.

J. J. More', B. S. Garbow, and K. E. Hillstrom [1981], "Testing unconstrained optimization software", *Transactions on Mathematical Software* 7, pp. 17-41.

J. J. More' and D. C. Sorensen [1982], "Newton's method", Argonne National Laboratory report ANL-82-8.

J. M. Ortega and W. C. Rheinboldt [1970], *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York.

L. B. Rall [1966], "Convergence of the Newton process to multiple solutions", *Numerische Mathematik* 9, pp. 23-37.

G. W. Reddien [1978], "On Newton's method for singular problems", *SIAM Journal on Numerical Analysis* 15, pp. 993-996.

G. W. Reddien [1980], "Newton's method and high order singularities", *Comput. Math. Appl.* 5, 79-86.

R. B. Schnabel [1977], "Analyzing and improving quasi-Newton methods for unconstrained optimization", Ph.D. thesis, Department of Computer Science, Cornell University.

R. B. Schnabel [1982a], "Unconstrained Optimization in 1981", in *Nonlinear Optimization 1981*, M. J. D. Powell, ed., Academic Press, London.

R. B. Schnabel [1982b], "Unconstrained minimization using conic models and second derivatives", to appear.

R. B, Schnabel and P. Frank [1982], "Tensor methods for solving systems of nonlinear equations", to appear.

D. C. Sorensen [1980], "The q-superlinear convergence of a collinear scaling algorithm for unconstrained optimization", *SIAM Journal on Numerical Analysis* 17, pp. 84-114.

K. A. Stordahl [1980], "Unconstrained minimization using conic models and exact second derivatives", M.S. thesis, Department of Computer Science, University of Colorado at Boulder.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER

4. TITLE *(and Subtitle)*

5. TYPE OF REPORT & PERIOD COVERED

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
U. S. Army Research Office
Post Office Box 12211
Research Triangle Park, NC 27709

12. REPORT DATE

13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*

15. SECURITY CLASS. *(of this report)*
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
NA

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

NA

18. SUPPLEMENTARY NOTES

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Conic methods, tensor methods, unconstrained minimization, nonlinear equations.

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*